# Assignment 02

Code the following in Python

The Google Finance API can be used to get stock price data for a given symbol at higher sampling rate than just daily frequency. In this example, YHOO prices from for 15 days with interval of 300 seconds are requested.

http://www.google.com/finance/getprices?q=YHOO&i=300&p=15d&f=d,o,h,l,c,v

In which the parameters are:

q= stock symbol

x= exchange symbol, may be omitted, for example the above can be

http://www.google.com/finance/getprices?q=YHOO&x=NASDAQ&i=300&p=15d&f=d,o,h,l,c,v

i = interval (here i=300 means intervals of 100 seconds or 5 minutes)

p = number of period (here 15d denotes 15 days of data)

f = parameters (day, close, open, high and low, volume)

The following is an excerpt of the data

```
EXCHANGE%3DNASDAQ
MARKET_OPEN_MINUTE=570
MARKET_CLOSE_MINUTE=960
INTERVAL=300
COLUMNS=DATE,CLOSE,HIGH,LOW,OPEN,VOLUME
DATA=
TIMEZONE_OFFSET=-300
a1452263400,30.49,30.51,30.49,30.51,133480
1,30.67,30.7,30.29,30.5,329140
2,30.75,30.782,30.61,30.67,338581
3,30.76,30.84,30.75,30.755,367287
4,30.86,30.8875,30.75,30.77,330992
5,30.815,30.9,30.81,30.86,404019
```

(3 points) Write a function that takes input parameters q, I, p, f, and sends requests to Google Finance API, retrieves the response, and returns the close price.

(2 point) Write code that reads a list of symbols from an Excel file, calls the function above for each stock symbol, returns data into a variable, and saves the data to an Excel file.

**Note:** If only close prices are needed, you will just need the parameter f=c, for example

http://www.google.com/finance/getprices?q=YHOO&i=300&p=15d&f=c

**Hint**: Use urllib2 to send requests to Google Finance API, some sample code is given below as hints. The url is like the link above with necessary stock symbol. Splitting using '\n' provides multiple rows, and splitting using ',' provides multiple columns for each row. If only f=c is given to the function, splitting with ',' is not necessary.

```
import urllib

…

response = urllib2.urlopen(url)

data = response.read().split('\n') # here split into multiple rows, data starts at index 7

for i in range(7, len(data)):

    # Now split each row into columns using the delimiter ',' – only if more than 1 column

….
```

***Submission includes code, all the Excel files, and a document explaining your work.***